Experience and Consideration in Safety Critical Software Tests

Do Young Oh*, Se Do Sohn, Sung Ho Kim, Chang Ho Kim, Woo Goon Kim, Kwon Ki Moon, Young Woo Chang
I&C System Engineering Department, Korea Power Engineering Co.
150 Duckjin-dong, Yuseong-gu, Daejeon 305-323
Corresponding author: grayo@kopec.co.kr

1. Introduction

Software for safety critical systems of nuclear power plants are developed based on a strict development process[1,2,3]. Software verification & validation (SW V&V) is a very important one among development activities[2]. Specially, testing part of SW V&V needs many efforts and know-how. KOPEC has experience from several projects for Nuclear Power Plants. Each SW V&V step of software development model has specific goals such as performance, calculation accuracy, response time, depending on test stage. To achieve each goal, various testing methodologies are adopted. When the results of each stage satisfy expected performance and accuracy, the next stage is performed. Faults are detected during tests, and if correcting faults needed software change, regression tests should be performed through all stages.

2. Tests classified by phases

The safety critical software is developed based on software life cycle models. KOPEC applies V-model to development process. So each test phase verifies and validates requirements corresponding to development phases.

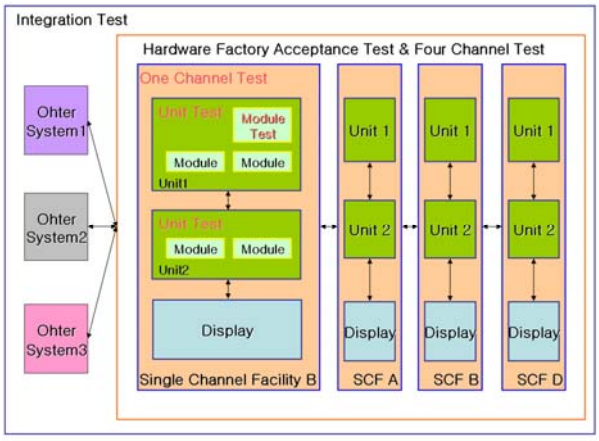

Figure 1. Test Scope

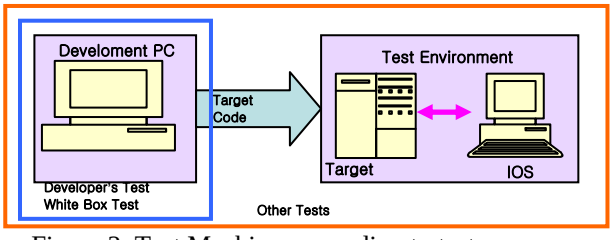

Figure 2. Test Machines according to test purposes

In the implementation phase the Developer's test is performed by Design Team. Its purpose is to confirm that functions of each module are correctly implemented manually or by automated method. The module test confirms correct operations of each module on target platform. Unit tests confirm every module is precisely connected and calculation results are correct in comparison with functional design. In the validation testing phase, the purpose is to confirm that intra-system or inter-system interfaces are precisely deployed and their operations are correct in accordance with requirement. Figure 1 and 2 show scope of each test.

3. Consideration in Implementation Phase

Before releasing codes to V&V team for test, developer need self-tests to verify implementation. The first verifying method is a code review and code walkthrough. The second method is logic tests on the developer's computer through just adding test interface logics. Because the test interface can be used commonly and Windows IDE supports powerful debugging functions, simple bugs which are developer's mistakes such as typo can be found and resolved. It helps to reduce burden of V&V testing. Figure 3 Shows developer's test IDE & its result.

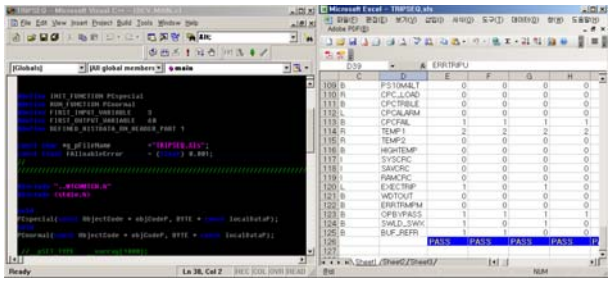

Figure 3. Development's Test Environment

4. Considerations in Module Tests

Module test consists of black box and white box tests. During the white box test, code coverage, correctness of implementation, and suitability of test case are verified. Through the black box test on target platform, correctness of implementation and calculation accuracy on the target platform is verified. However, modules are just set of logics and expressions implemented by ANSI-C, it is necessary to write interface logics. Manual adding test I/O interfaces depending on hundreds of modules is not efficient and may cause human error, so automated code generation is preferable.

4.1. White box test Automation

White box test is conducted with LDRA Testbed. LDRA testbed is a specialized software tool for test, and it supports great parts of automation to generate interface codes. But a tester has to add a part loading test case. These codes are generated automatically with perl script.

4.2. Considerations of black box test

Because Black box test is performed on Single Channel Facility (SCF, Figure 4), there are two big considerations during Black Box Tests. One comes from test method via communication. In case that one data type is supported for communication, writing interface blocks is a hard work. It can be resolved by generating that blocks automatically through backtracking of test case file. The other is data format problem due to different processor. In case that IO Simulator(IOS) and target machine use different data format, IOS has to convert its data format into data format of target machine before sending and after receiving data.

Figure 4. Single Channel Facility(SCF)

5. Unit Tests (Integration Testing)

Unit Tests are composed of Input Sweep Test(IST), Dynamic Test(DT), and Live Input Test(LIT). IOS provides test cases for IST and DT via communication and for LIT via signals to I/O cards. IOS receives test results via communication for all three tests. These tests are performed on the SCF (Figure 4). Except for module test, there are several aspects to require consideration to perform suitable unit test.

- Timing consideration (DT, LIT)
- Auto restart function of target machine by IOS(IST, DT)
- enable/completion processing(IST, DT, LIT) of testing
- Test harness (IST, DT)

Timing is very important in Unit Tests. DT and LIT need dynamic changes of communication data. To meet timing requirements, communication via high speed link is absolutely necessary. IST and DT perform more than one thousand test cases and each test case takes several minutes. Automatic restart of target machine by IOS helps to reduce human efforts, errors, and required time. All tests start after a set of data is prepared. LIT and DT demand steady state of algorithms to begin tests. IOS needs to receive flags that mean preparation or completion of test via communication. To perform these tests, tester changes or adds additional codes depending on demands.

6. Validation Testing

Safety critical systems are composed of four channels and One Channel System Test validates whether one of them correctly perform its function. This test includes display, data link, system fault, system failure response, and etc. Hardware Factory Acceptance Test verifies proper assembly, and wiring of system deliverable equipment covers the necessary items such as I/O terminal block string check, calibration AI card, verifying communication links, and so on. IT is the hardware and software integration tests with other interfaced systems and checks inputs correctly processed and displayed, outputs correctly presented, and interface with other systems properly operated. Because the main purposes of these tests are to confirm that system is configured and interface with other systems, information is correctly displayed. So there is little opportunity to perform test more efficiently.

6. Conclusion

During safety critical software development process, various tests are required. Because they demand much time and other resources, reducing errors at the early stages and setting up an automatic test environment is necessary. To generate accurate software with limited resources and time, the first test of V&V tasks (i.e., Module test) is the most important one among these tests. That is because if error is detected at the IT phase and it needs a change of module code, all tests are re-performed and they require huge time and efforts. Test automation helps testers reduce efforts and human errors and concentrate themselves on test. Detection of errors at early phase in module test can reduce times and efforts. This way, more reliable software can be produced efficiently.

REFERENCES

[1] IEEE Std 7-4.3.2-2003, IEEE Standard Criteria for Digital Computers in Safety systems of Nuclear Power Generating Stations.
[2] IEEE Std 1012-2004, IEEE Standard for Software Verification and Validation.
[3] IEEE Std 1074-2006, IEEE Standard for Developing Software Life Cycle Processes.